# AppleScript 도움말

필수 항목

## AppleScript에 관하여

Mac을 사용할 때 반복해서 하는 작업이 있을 수 있습니다. AppleScript를 사용하여 반복되는 작업을 자동화할 수 있는 스크립트와 응용 프로그램을 작성할 수 있습니다.

AppleScript는 OS X에 내장되어 있는 강력하고 실용적인 스크립팅 언어입니다. AppleScript를 사용하여 단축키 생성, 반복 작업 자동화가 가능하며 상당한 시간을 절약하는 응용 프로그램 사용자화도 가능합니다. AppleScript를 사용하여 스크립트 가능한 응용 프로그램을 제어할 수 있습니다.

AppleScript 스크립트는 객체에 동작을 수행하는 영어와 같은 명령어로 구성됩니다. 동작의 몇가지 예로 열기, 닫기 또는 이름 가져오기가 있습니다. 응용 프로그램, 윈도우 또는 환경설정 설정값과 같은 것이 대상체입니다.

스크립트는 하나 이상의 응용 프로그램을 조정할 수 있습니다. 스크립트가 가능하려면 응용 프로그램은 스크립트에서 사용할 수 있는 용어를 정의해야 합니다. OS X은 스크립트 가능한 여러 응용 프로그램을 제공합니다.

스크립트는 네트워크 상에서 응용 프로그램을 조정할 수도 있으므로 1대 이상의 컴퓨터에서 작업을 자동화할 수 있습니다.

사실, 여러분은 이러한 사실을 알지 못하고 AppleScript를 사용해 왔습니다. 시스템 및 응용 프로그램의 여러 기능은 응용 프로그램 내에서 호출된 실제 AppleScript 스크립트입니다. 예를 들어, 이 도움말 시스템에서 응용 프로그램이나 파일을 여는 링크를 클릭할 때마다 AppleScript 스크립트가 동작을 수행합니다.

**관련 주제**

AppleScript 작동 방법
AppleScript 편집기에 관하여
AppleScript 언어에 관하여
인터넷 상의 AppleScript 리소스
AppleScript Studio 개발 환경

필수 항목

## AppleScript 편집기에 관하여

AppleScript 편집기는 AppleScript 스크립트를 읽고, 쓰고, 기록하고 저장할 수 있는 쉽게 사용할 수 있는 스크립트 응용 프로그램입니다.

다음은 AppleScript 편집기 기능의 일부입니다.

- 통합되고 사용하기 쉬운 인터페이스를 제공합니다.
- 스크립트에서 텍스트를 찾고 대치할 수 있습니다.
- 긴 스크립트 구문을 자동으로 줄 바꾸기할 수 있습니다.
- 여러 개를 취소할 수 있습니다.
- 보다 빠른 스크립트 작성을 위해 단축키 메뉴 스크립트를 제공합니다.
- 응용 프로그램 서비스를 지원합니다.

- 라이브러리 팔레트를 사용하여 응용 프로그램 스크립트에 접근할 수 있습니다.
- 구문을 완성하는 것을 돕기 위해 스크립트 지원을 제공합니다.
- 결과 히스토리와 이벤트 로그 히스토리를 제공하여 스크립트에서의 진행을 검사할 수 있습니다.

**관련 주제**

AppleScript 언어에 관하여
AppleScript Studio 개발 환경

---

필수 항목

## AppleScript 스크립트 생성하기

AppleScript를 사용하기 위해 프로그래밍에 대한 모든 것을 알 필요는 없습니다. 사실, 스크립트의 한 단어도 작성할 필요가 없습니다. 유용하게 사용할 수 있는 수 많은 AppleScript 스크립트가 Mac OS X에 포함되어 있으며 이 보다 더 많은 스크립트를 웹에서 구할 수 있습니다.

조금 더 깊게 들어가고 싶은 경우 기존의 스크립트를 자신의 필요에 맞게 약간 수정할 수 있습니다.

물론, 자신만의 스크립트를 작성한다면 보다 유용할 것입니다. 스크립트 작성 방법을 배우면 유용한 도구 및 시간 절약을 위한 자동화 도구를 생성하여 작업을 보다 생산적으로 할 수 있습니다.

스크립트를 작성할 수 있는 간단한 환경을 제공하는 AppleScript 편집기를 사용하여 자신만의 스크립트를 수정하거나 생성할 수 있습니다. AppleScript 도움말의 모든 예제는 AppleScript 편집기를 사용하여 작성되었습니다.

---

- AppleScript 편집기를 엽니다.

  AppleScript 편집기 윈도우에 다음 텍스트를 입력하십시오.

  ```
  say "Welcome to AppleScript!"
  ```

  실행을 클릭하십시오.

---

**관련 주제**

AppleScript 편집기에 관하여
AppleScript Studio 개발 환경

---

필수 항목

## AppleScript 작동 방법

AppleScript는 “Apple 이벤트”라는 메시지를 보내 동작합니다. 스크립트 실행 시 AppleScript는 Apple 이벤트를 명령을 실행하는 시스템이나 응용 프로그램에 보냅니다.

대부분의 경우, 응용 프로그램이나 시스템은 결과를 AppleScript에 반환합니다. 예를 들어 스크립트가 요청을 응용 프로그램에 보내 대상체의 이름을 얻은 경우 응용 프로그램은 이름을 반환합니다. 명령을 수행할 수 없는 경우 응용 프로그램이나 AppleScript가 오류 메시지를 반환할 수 있습니다. AppleScript 편집기에서 반환된 결과를 볼 수 있습니다.

일반적으로 Apple 이벤트의 메커니즘에 대해 알 필요는 없습니다. AppleScript는 백그라운드에서 이 모든 것을 수행합니다. 스크립트를 작성하고 실행하기만 하면 됩니다.

**관련 주제**

필수 항목

## 애플릿에 관하여

응용 프로그램으로 저장된 스크립트를 AppleScript 응용 프로그램 또는 애플릿이라고 합니다. 애플릿을 열면 자동으로 실행되어 스크립트의 명령을 수행합니다. 애플릿은 단지 두서너 개의 명령을 포함한 작은 함수이거나 수천 행을 포함한 복잡한 응용 프로그램일 수 있습니다.

애플릿을 편집 가능하거나 실행 전용으로 저장할 수 있습니다. 편집 가능한 애플릿은 스크립트 텍스트를 포함하여 언제든지 편집하여 다시 컴파일할 수 있습니다. 애플릿을 편집하거나 변경하려면, AppleScript 편집기 아이콘 위로 드래그하십시오. 스크립트를 실행 전용으로 저장하면 편집 가능한 텍스트 부분이 삭제되어 파일을 더이상 편집하거나 재컴파일할 수 없습니다.

편집 가능하도록 저장한 애플릿을 편집하려면 AppleScript 편집기나 다른 AppleScript 편집 응용 프로그램에서 여십시오. 애플릿은 해당 애플릿을 열 때 스크립트 실행을 시작하는 "on run" 핸들러가 필요합니다.

- 다음 예제는 "on run" 핸들러를 보여줍니다.

```
on run
        --first command
        --second command
end run
```

**관련 주제**

드롭릿에 관하여
AppleScript 파일 유형

필수 항목

## 컴파일된 스크립트에 관하여

컴파일된 스크립트는 AppleScript 편집기나 다른 AppleScript 편집 프로그램에서 읽고 편집할 수 있는 편집 가능한 형식으로 된 AppleScript 파일입니다.

컴파일된 스크립트를 열면, AppleScript 편집기에 나타나지만 명령을 자동으로 실행하지는 않습니다. 따라서 실제로 실행하지 않고도 스크립트를 변경할 수 있습니다. 파일 내에서 모든 텍스트를 읽고 AppleScript 코드로 변환하는 스크립트를 컴파일할 수 있습니다. 컴파일된 스크립트를 실행할 수 있습니다. 컴파일된 스크립트의 파일 확장자는 .scp입니다.

파일을 컴파일된 스크립트로 저장해도, 텍스트는 코드와 함께 파일에 남습니다. 그렇기 때문에 언제든지 컴파일된 스크립트를 다시 편집할 수 있습니다.

**관련 주제**

애플릿에 관하여
드롭릿에 관하여
AppleScript 파일 유형

필수 항목

## 드롭릿에 관하여

드롭릿은 파일이나 폴더를 아이콘에 드래그시 실행되는 AppleScript 스크립트입니다. 스크립트는 파일이나 폴더에 대해 동작을 수행합니다.

애플릿과 마찬가지로 드롭릿을 편집 가능하거나 실행 전용으로 저장할 수 있습니다. 편집 가능한 드롭릿은 스크립트 텍스트를 포함하여 언제든지 편집하여 다시 컴파일할 수 있습니다. 드롭릿을 편집하거나 변경하려면 스크립트 편집기 아이콘 위로 드래그하십시오. 스크립트를 실행 전용으로 저장하면 편집 가능한 텍스트 부분이 삭제되어 파일을 더이상 편집하거나 재컴파일할 수 없습니다.

드롭릿을 열 때 스크립트 실행을 시작하는 “on open” 핸들러가 필요합니다. “on open” 핸들러에는 드롭릿에 드래그한 항목의 목록을 포함하는 매개변수가 있습니다. 드롭릿 스크립트의 일부는 목록을 검사하고 다른 부분은 항목에 동작합니다. 드롭릿이 단일 항목으로만 동작하도록 제작되지 않았다면 목록에 있는 항목이 교대로 동작을 수행하는 동안 반복되는 반복 루프를 포함해야 합니다.

---

- 다음은 “on open” 및 매개변수 목록을 포함한 드롭릿 핸들러의 예제입니다.

```
on open (theItemsDropped)
        --first command
        --second command
end open
```

---

**관련 주제**

애플릿에 관하여
컴파일된 스크립트에 관하여
AppleScript 파일 유형

---

필수 항목

## AppleScript 사전 사용하기

응용 프로그램을 조정할 수 있는 명령어와 대상체를 찾으려면 응용 프로그램의 사전을 보십시오. 사전에는 응용 프로그램에서 이해하는 용어 세트가 포함되어 있습니다.

라이브러리 윈도우에서 응용 프로그램을 선택하거나 응용 프로그램의 아이콘을 AppleScript 편집기로 드래그하면 AppleScript 편집기에서 응용 프로그램의 사전을 볼 수 있습니다.

사전에 있는 용어는 “모음(Suite)”라고 불리는 관련 용어의 그룹으로 구성되어 있습니다. 명령과 대상체는 볼드체 텍스트입니다.

---

1. AppleScript 편집기에서 윈도우 > 라이브러리를 선택하십시오.
2. 라이브러리 윈도우에서 응용 프로그램 이름을 이중 클릭하십시오.
3. 모음을 선택하십시오.

   해당 모음의 용어 및 설명이 나타납니다.

---

AppleScript 언어는 다양한 카테고리로 정렬되어 있습니다.

- 모음: 관련된 용어 그룹. AppleScript 편집기 사전에서 모음은 첫 번째 열에 나타나고 주황색 문자 "S"로 표시됩니다.
- 명령: 동작이나 결과 요청. “이벤트"라고도 하는 명령이 두 번째 열에 나타나고 파란 색 문자 "C"로 표시됩니다(이벤트라고도 함).
- 클래스: 스크립트 가능한 대상체. 클래스 설명은 해당 클래스에 있는 속성 및 요소를 나열합니다. 클래스는 두 번째 열에 나타나며 엷은 자주색 “C”로 표시됩니다.

- 속성: 값을 포함하는 대상체. 속성는 세 번째 열에 나타나며 엷은 자주색 “P”로 표시됩니다.
- 요소: 다른 대상체에 포함된 대상체. 예를 들어 단락에는 여러 단어가 포함될 수 있습니다. 요소는 세 번째 열에 나타나며 주황색 문자 “E”로 표시됩니다.

**관련 주제**

클래스 정의
명령어 정의
응용 프로그램 명령에 AppleScript 사용하기

**열기**

AppleScript 언어 설명서: 명령 참조
AppleScript 언어 설명서: 클래스 참조

---

필수 항목

## 문제 해결하기

AppleScript를 사용하는 데 문제가 발생하는 경우 문제를 해결할 수 있는 정보를 위해 다음 주제를 탐색하십시오.

Choose From List 대화상자에서 취소를 클릭하면 오류가 발생합니다

폴더 적용 스크립트가 예기치 않게 동작하는 경우

스크립트에서 디스크가 인식되지 않는 경우

---

기본 명령어

## 클래스 정의

클래스 정의는 스크립트 가능한 대상체의 특성 및 요소를 설명합니다. 클래스는 문장에서의 명사에 비유할 수 있으며 클래스 정의는 명사의 사전 정의와 같습니다.

보통, AppleScript 사전에 있는 클래스 정의는 볼드체로 시작되고 같은 줄에 용어의 짧은 설명이 제공됩니다.

그 뒤에 스크립트 가능한 대상체에 속하거나 구성하는 요소나 대상체 목록이 나타납니다. 일부 스크립트 가능한 대상체는 다른 스크립트 가능한 대상체만을 포함하고 요소를 포함하지 않기 때문에 요소가 항상 모든 클래스 정의에 포함되는 것은 아닙니다.

클래스 정의는 스크립트 가능한 대상체의 속성이나 크기 및 값의 목록으로 끝납니다.

속성 엔트리는 속성의 이름으로 시작하며 다음 내용이 이어집니다.

- 속성 값이 저장될 데이터 포맷의 정의. 예: integer(숫자), point(두 개의 숫자 목록), string(텍스트 데이터) 및 Boolean(참 또는 거짓)
- 속성의 값이 “읽기 전용”임을 나타내는 선택적 유형 표시자 [r/o](읽을 수는 있지만 변경할 수 없음)
- 속성에 관한 짧은 설명

대상체가 복수 형태인 경우 하나 이상의 대상체를 참조할 수 있습니다.

대상체에 요소 목록이 있는 경우 다른 대상체를 포함합니다. 해당 스크립트에서 사용할 수 있는 요소 이름은 볼드체로 나타납니다. 보통체의 텍스트는 요소 참조하는 방법을 설명합니다.

대상체는 속성을 가질 수 있습니다. 속성의 이름은 볼드체로 나타납니다. 보통체의 텍스트는 문자열과 같이 속성을 포함한 값의 종류를 설명합

니다. 속성은 상수 값이나 상수 목록 중 하나일 수 있습니다. 구문 다음에 속성이 [r/o](읽기 전용)이 아니라면 스크립트는 속성 값을 얻을 수 있고, 설정할 수 있습니다.

**관련 주제**

명령어 정의
대상체 정의
AppleScript 명령
AppleScript 사전 사용하기

**열기**

AppleScript 언어 설명서: 클래스 참조

---

기본 명령어

## 명령어 정의

명령어(Command)는 동작을 수행하는 AppleScript에서의 용어들입니다. AppleScript 사전에서 명령어는 가운데 열에 나타나며 파란색 문자 "C"로 표시됩니다. 명령어 정의는 명령어와 구문이 무엇을 실행할 지에 관한 간단한 설명을 포함합니다. 구문은 스크립트에서 명령어를 쓰는 방법입니다. 구문은 여러 개의 부분으로 구성될 수도 있습니다.

볼드체의 단어는 명령문에서 사용하는 용어입니다. 사전에서 나타나는 것과 일치하게 용어를 써야 합니다.

파란색의 텍스트 단어는 사용자가 제공한 정보를 말합니다. 예를 들어, 사용자가 정한 대상체나 변수를 지정할 수 있습니다.

명령어는 때때로 괄호 안에 나타나는 선택사항 부분을 가지고 있습니다. 구문에 사용할 필요는 없습니다.

**관련 주제**

클래스 정의
AppleScript 사전 사용하기
이벤트 정의
대상체 정의

**열기**

AppleScript 언어 설명서: 명령 참조

---

기본 명령어

## 이벤트 정의

이벤트 정의는 스크립트 가능한 대상체에서의 동작에 사용하는 동사 또는 명령을 설명합니다. 이벤트는 대상체가 수행하거나 참여할 동작이나 동사입니다.

보통, 이벤트 정의는 용어가 나오고 같은 행에 용어의 짧은 설명이 제공됩니다.

그 다음에는 명령 실행에 사용하는 매개변수 목록이 나옵니다. 선택사항인 매개변수가 중괄호 안에 나타납니다. 매개변수가 꼭 필요한 것은 아닙니다.

각 매개변수 엔트리는 매개변수 이름으로 시작되며, 매개변수에서 사용하는 데이터 포맷 설명이 나오고, 매개변수에 관한 짧은 설명으로 끝납니다.

명령어 정의는 명령어와 구문이 무엇을 실행할 지에 관한 간단한 설명(밑줄 친 텍스트)을 포함합니다. 구문은 스크립트에서 명령어를 쓰는 방법입니다. 구문은 여러 개의 부분으로 구성될 수도 있습니다.

볼드체의 단어는 명령문에서 사용하는 용어입니다. 사전에서 나타나는 것과 일치하게 용어를 써야 합니다.

보통체의 텍스트 단어는 사용자가 제공하는 정보를 말합니다. 예를 들어, 사용자가 정한 대상체나 변수를 지정할 수 있습니다.

명령은 때때로 괄호 안에 나타나는 선택사항 부분을 가지고 있습니다. 구문에 사용할 필요는 없습니다.

명령은 대상체에 따라 동작합니다. 대상체 정의는 이름과 간단한 설명(밑줄친 텍스트)을 포함한, 대상체를 지정하는 구문을 나타냅니다. 대상체의 클래스는 대상체와 함께 사용할 수 있는 명령어와 같은 대상체의 특성을 결정합니다.

**관련 주제**

데이터베이스 이벤트
클래스 정의
명령어 정의
대상체 정의
AppleScript 작동 방법

---

기본 명령어

## 대상체 정의

명령어는 대상체에 따라 동작합니다. AppleScript 사전에 있는 대상체 정의는 이름과 간단한 설명(밑줄친 텍스트)을 포함한 대상체를 지정하는 구문을 나타냅니다. 대상체의 클래스는 대상체와 함께 사용할 수 있는 명령어와 같은 대상체의 특성을 결정합니다.

대상체가 복수 형태인 경우 하나 이상의 대상체를 참조할 수 있습니다.

대상체에 요소 목록이 있는 경우 다른 대상체를 포함합니다. 해당 스크립트에서 사용할 수 있는 요소 이름은 볼드체로 나타납니다. 보통체의 텍스트는 요소 참조하는 방법을 설명합니다.

대상체는 속성을 가질 수 있습니다. 속성의 이름은 파란색 볼드체로 나타납니다. 파란색 보통체의 텍스트는 문자열과 같이 속성을 포함한 값의 종류를 설명합니다. 속성은 상수 값이나 상수 목록 중 하나일 수 있습니다. 구문 다음에 속성이 "[r/o]"(읽기 전용)가 아닌 경우 스크립트는 속성 값을 얻을 수 있고 설정할 수 있습니다.

**관련 주제**

응용 프로그램 명령에 AppleScript 사용하기
명령어 정의
클래스 정의
대상체 동작 말하기
AppleScript 사전 사용하기

**열기**

AppleScript 언어 설명서: 클래스 참조
AppleScript 언어 설명서: 속성

---

기본 명령어

## If 문

If 문은 특정 조건을 만족할 때만 실행되는 명령입니다. 각 if 문에는 참인지 거짓인지를 평가하는 부울 표현식을 포함합니다. if 문 내에 포함된 명령은 부울 표현식의 값이 참(true)일 경우에만 실행됩니다. if 문은 end if로 끝납니다. 예:

```
if amountReceived > 100 then
        set memberStatus to "special"
```

```
end if
```

위의 구문에서 부울 표현식이 참이면(amountReceived가 100보다 큰 경우) if 문 내의 명령이 실행됩니다. 부울 표현식이 거짓(false)인 경우 명령은 실행되지 않습니다.

if 문이 else 문과 함께 구성될 수도 있습니다. 이 조합은 if-then-else 문으로 알려져 있습니다. 예:

```
if amountReceived > 100 then
        set memberStatus to "special"
else
        set memberStatus to "general"
end if
```

위의 구문에서 첫 번째 부울 표현식이 참인 경우(받은 금액이 100보다 큰 경우) 구문 첫 번째 부분의 명령이 실행됩니다. 첫 번째 부울 표현식이 거짓인 경우 else 뒤의 두 번째 부분의 명령이 실행됩니다.

다중 조건을 처리하기 위해 if 문을 다른 if 문 안에서 중첩해서 사용할 수 있습니다.

AppleScript의 if 문에 대해 더 자세히 알아보려면 AppleScript 언어 설명서를 참조하십시오.

**관련 주제**

반복 구문
명령어 정의
AppleScript에서 변수 설정하기
상수 값

**열기**

AppleScript 언어 설명서 - If

---



# 반복 구문

조건에 따라 작업을 여러 번 수행하기 위해 repeat 문을 사용할 수 있습니다. repeat 문은 조건이 만족되고 루프가 종료될 때까지 루프에 있는 구문의 그룹을 실행합니다. 다양한 조건에 응답하기 위해 repeat 문을 다른 repeat 문 안에 겹쳐 넣을 수 있습니다.

AppleScript에는 다른 조건에 응답할 수 있는 다양한 repeat 문이 있습니다.

### repeat (계속)

루프에서 계속 반복합니다. exit 문은 루프에서 나갈 때 사용하십시오.

```
repeat
        -- statement
        -- statement
        if (x = 5) then exit repeat
end repeat
```

### repeat (횟수) times

지정된 횟수 동안 반복합니다.

```
repeat 5 times
	-- statement
	-- statement
end repeat
```

---

**repeat while**

지정된 조건이 참인 동안 반복합니다.

```
set x to 1
repeat while (x < 5)
	-- statement
	-- statement
	set x to (x + 1)
end repeat
```

---

**repeat until**

지정된 조건이 참일 때까지 반복합니다.

```
set x to 1
repeat until (x = 5)
	-- statement
	-- statement
	set x to (x + 1)
end repeat
```

---

**repeat with (변수) from (시작값) to (중단값)**

변수에 시작값를 설정한 다음 루프를 돌 때마다 변수를 증가시킵니다. 변수가 중단값에 도달하면 repeat 문이 끝납니다.

```
repeat with x from (5) to (25)
	-- statement
	-- statement
end repeat
```

---

**repeat with (변수) in (목록)**

변수에 목록의 첫 번째 항목을 설정합니다. 루프를 돌 때 변수는 다음 항목으로 설정됩니다. 목록의 모든 항목이 사용되면 repeat가 끝납니다.

```
set myList to {234, 342, 681, 874}
repeat with x in myList
	-- statement
	-- statement
end repeat
```

---

AppleScript의 repeat 문에 대해 더 자세히 배우려면 AppleScript 언어 설명서를 참조하십시오.

**관련 주제**

if 구문

AppleScript에서 변수 설정하기
상수 값

**열기**

AppleScript 언어 설명서 - repeat

---

기본 명령어

## 상수 값

상수는 이미 지정된 값을 포함하는 AppleScript의 예약된 단어입니다. AppleScript에는 대상체에 관한 정보를 얻거나 비교하거나 또는 수학적 계산을 하는데 사용할 수 있는 여러 상수가 포함되어 있습니다. 응용 프로그램과 기타 대상체는 자신만의 상수를 정의할 수 있습니다. 이들 상수들은 value 클래스 정의에서 정의됩니다.

상수 값을 보려면 다음 예제를 AppleScript 편집기에 붙이기한 다음 실행을 클릭하십시오. 그런 다음 결과를 눌러 상수 값을 보십시오.

결과 또한 상수입니다. 명령에서 반환된 값을 참조하는 스크립트 상수입니다.

```
get the current date
```

다음은 상수의 몇 가지 다른 예제입니다.

- 연산 상수: pi
- 부울 상수: true, false
- 날짜 및 시간 상수: minutes, hours, days, weeks
- 문자열 상수: return, space, tab
- 스크립트 상수: anything, current application, it, me, missing value, my, result

AppleScript의 상수에 대해 더 자세히 알아보려면 AppleScript 언어 설명서를 참조하십시오.

**관련 주제**

AppleScript에서 변수 설정하기
대상체 정의
클래스 정의
이벤트 정의

**열기**

AppleScript 언어 설명서 - constants

---

기본 명령어

## AppleScript에서 변수 설정하기

변수는 값을 저장하는데 사용되는 스크립트 내에 생성된 컨테이너입니다. 변수를 생성할 때 ID라고도 불리는 변수에 대한 이름을 지정합니다. 스크립트 실행 시, 값을 변수에 넣거나 값을 이용하거나 수정할 수 있습니다. 스크립트를 싱행하는 동안에만 변수가 유지됩니다. 스크립트를 종료하는 시점에서 값은 유실됩니다.

---

■ 다음 예제들은 고유 ID를 지정하고 값을 이에 할당하는 방법을 보여줍니다.

```
set myVar to "153"

set x to "Quarterly Reports"

set x to false
```

변수에 저장된 값이 무엇인지 보려면 이 예제들을 AppleScript 편집기에 붙이기한 다음 실행을 클릭하십시오. 결과를 클릭하여 변수에 저장된 값을 보십시오.

AppleScript의 변수에 대해 더 자세히 알아보려면 AppleScript 언어 설명서를 참조하십시오.

**관련 주제**

상수 값
가정문
반복 구문

**열기**

AppleScript 언어 설명서 - 변수

---

기본 명령어

## 응용 프로그램의 위치 지정하기

OS X에서 응용 프로그램은 "패키지"로 묶여 있습니다. 패키지는 응용 프로그램을 구성하는 폴더와 파일을 포함하는 특별한 디렉토리입니다. 이 때문에 "path to"와 "info for" 명령은 응용 프로그램에 대한 다른 결과를 반환할 수 있습니다.

응용 프로그램에 대해 "path to" 명령을 사용하는 경우 경로는 주로 ".app"으로 끝납니다. 파일 확장자는 파일 이름의 일부로 Finder에서 표시되지 않지만 스크립트에서 파일을 표시하기 위해 확장자를 사용할 수 있습니다.

응용 프로그램에 대한 정보를 얻기 위해 "info for" 명령을 사용하는 경우 결과의 Finder 속성은 주로 "true"입니다. 확장자 가리기, 이름 확장자 또는 표시된 이름과 같은 결과의 추가 속성을 사용하여 파일과 이름에 대한 더 많은 정보를 얻을 수 있습니다.

**관련 주제**

path to
info for
폴더, 파일 및 디스크 참조하기
choose application
Finder 스크립팅 명령

---

Finder 명령

## 폴더, 파일 및 디스크 참조하기

AppleScript로 경로를 포함하지 않고 이름으로 특정 폴더, 파일 및 디스크를 참조할 수 있습니다. AppleScript는 도큐멘트 폴더, 데스크탑, 홈 폴더 및 시동 디스크와 같은 항목들의 위치를 인식할 수 있습니다.

---

- 다음 예제는 스크립트에서 해당 이름을 사용하여 특정 폴더를 참조하는 방법을 나타냅니다.

```
set x to the path to the desktop folder
```

**관련 주제**

스크립트에서 Finder 참조하기
path to
info for
Finder 스크립팅 명령



## 스크립트에서 Finder 참조하기

Finder 명령을 사용하려면 명령을 응용 프로그램을 지시할 때와 같이 tell 문의 Finder에 명령을 지시해야 합니다.

- 다음은 스크립트에서 tell 문을 사용한 예제입니다. tell 문은 다양한 윈도우 기능을 “Finder.app”에 지시합니다.

```
tell application "Finder.app"
        return the number of windows
end tell
```

**관련 주제**

Finder
대상체 동작 말하기
폴더, 파일 및 디스크 참조하기
대상체 정의
Finder 스크립팅 명령

**열기**

AppleScript 언어 설명서 - 대상체 지정자



## Finder 스크립팅 명령

다음은 Finder와 함께 사용할 수 있는 명령입니다.

activate

alias file

alias list

application file

clean up

close

column

Computer-object

container

copy

count

data size

delete

desktop window

desktop

disk

document file

duplicate

eject

empty trash

exists

file

file information

find

Finder

Finder window

folder

get

icon view options

information window

Internet location file

item

list view options

make

move

moving folder window for

open

package

path to

preferences

preferences window

print

quit

폴더, 파일 및 디스크 참조하기

스크립트에서 Finder 참조하기

reveal

select

set

sort

trash

update

window

---



# 표준 스크립팅 추가 명령

스크립팅 추가는 사용자의 스크립트에 포함할 수 있는 유용하게 사용되는 다양한 명령을 추가합니다.

adding folder items to

beep

choose application

choose color

choose file

choose file name

choose folder

choose from list

choose URL

clipboard info

close access

closing folder window for

current date

delay

dialog reply

display

display dialog

do shell script

download

FTP item

get eof

info for

list disks

list folder

load script

mount volume

offset

open for access

open location

opening folder

path to

POSIX path

random number

read

round

run script

say

scripting components

set eof

set the clipboard to

set volume

store script

summarize

system attribute

time to GMT

upload

URL

write